#19
KW8
1 of 3

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants:	James Q. Mi, et al.	§	Group Art Unit:	2132
		§		
Serial No.:	09/259,620	§		
		§	Examiner:	Douglas J. Meislahn
Filed:	February 26, 1999	§		
		§		
For:	COMPUTER SYSTEM IDENTIFICATION	§ §	Atty. Dkt. No.:	ITL.0160US

Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

APPEAL BRIEF

Dear Sir:

Applicant hereby appeals from the Final Rejection dated March 31, 2003, finally rejecting claims 1, 3-6 and 8-38.

I. REAL PARTY IN INTEREST

The real party in interest is Intel Corporation, the assignee of the present application by virtue of the assignment recorded at Reel/Frame 9797/0271.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

Date of Deposit: July 9, 2003
I hereby certify under 37 CFR 1.8(a) that this correspondence is being deposited with the United States Postal Service as **first class mail** with sufficient postage on the date indicated above and is addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Janice Munoz

Janice Munoz

III. STATUS OF THE CLAIMS

Claims 1, 3-6 and 8-38 have been finally rejected and are the subject of this appeal.

IV. STATUS OF AMENDMENTS

An amendment is being concurrently filed herewith to correct a typographical error in claim 27. Because this amendment further narrows down the issues on appeal, it is assumed that this amendment will be entered. There are no other unentered amendments.

V. SUMMARY OF THE INVENTION

Referring to Fig. 1, an embodiment 10 of a computer system in accordance with the invention includes an encryption unit 31 that may receive identification requests from web sites 36 (web sites 36a, 36b and 36c, as examples) for an identity of the computer system 10. In response to these requests, the encryption unit 31 may furnish different hash values 32 (hash values 32a, 32b and 32c, as examples) to the different web sites 36. In some embodiments, each hash value 32 is different, and as a result, each web site 36 may identify the computer system 10 by a different hash value 32, although each of the hash values 32 is generated by a single processor number 30, as described below. Because each web site 36 associates the computer system 10 with a different hash value 32, information about a user of the computer system 10 may not be correlated between databases that are maintained by different web sites 36. For example, a particular web site 36 may identify the computer system 10 via the hash value "1bdf23" and another web site 36 may identify the computer system 10 via the hash value "53gh44." Furthermore, as described below, the manner in which the encryption unit 31 generates the hash values 32 makes it very difficult for a rogue web site 36 from obtaining the hash values 32 that identify the computer system 10 to other web sites 36. Therefore, due to the technique used by the encryption unit 31, it may be very different to correlate information about the user from databases that are maintained by different web sites 36. In this context, the term "web site" generally refers to an arrangement where a computer system (a server, for example) executes software to provide a service to other computer systems, such as the computer system 10. Specification, pp. 3-4.

In the context of this application, the phrase "computer system" may generally refer to a processor-based system and may include (but is not limited to) a graphics system, a desktop computer, a mobile computer (a laptop computer, for example), or a set-top box as just a few examples. The term "processor" may refer to, as examples, at least one central processing unit (CPU), microcontroller, X86 microprocessor, Advanced RISC Machine (ARM) microprocessor or Pentium-based microprocessor. The examples listed above are not intended to be limiting, but rather, other types of computer systems and other types of processors may be included in some embodiments of the invention. Specification, p. 4.

To obtain a hash value 32 that identifies the computer system 10, a particular web site 36 may transmit a privacy key 34 (privacy keys 34a, 34b and 34c, as examples) to the computer system 10. In response, the encryption unit 31 may encrypt an embedded identifier, such as a processor number 30, with the privacy key 34 to produce the hash value 32 that the computer system 10 furnishes to the requesting web site 36. In this manner, if each web site 36 transmits a different privacy key 34 to the computer system 10, then each web site 36 receives a different hash value 32, each of which indicates the computer system 10 to the particular web site 36. As described further below, the encryption unit 31 may include a processor 200 (see Fig. 3) to aid in the encryption of the privacy key 34 with the processor number 30. Specification, pp. 4-5.

The privacy key 34 may or may not be a private key, depending on the particular embodiment. For example, in some embodiments, the privacy key 34 may be derived from an address or universal resource locator (URL) for the web site 36. Therefore, as an example, the privacy key 34 may indicate a string, such as "www.example.com." As described below, for the embodiments where the privacy key 34 is derived from the URL, the computer system 10 may perform a validity check to determine if the privacy key 34 that is furnished by a particular web site 36 is based on the URL of the web site 36. Specification, p. 5.

In some embodiments, the encryption unit 31 may use a hash function called F(PN, PRIVACYKEY) to perform the encryption. The F(PN, PRIVACYKEY) function may have properties that make it more difficult to track user information (about the computer system 10) that is stored on different web sites 36. For the F(PN, PRIVACYKEY) hash function, the notation "PN" represents the processor number 30, and the notation "PRIVACYKEY" represents the privacy key 34. Specification, p. 5.

One of the properties of the F(PN, PRIVACYKEY) hash function may be that the F(PN, PRIVACYKEY) function is a one way hash function, a notation that implies given the hash value 32 and the privacy key 34, it may be very difficult, if not impossible, to work backwards to determine the processor number 30. As a result, it may be very difficult for a particular web site 36 to use the hash value 32 that is obtained by that web site 36 to derive the processor number 30. Specification, p. 5.

In some embodiments, another property of the F(PN, PRIVACYKEY) function may be that the F(PN, PRIVACYKEY) function is collision free, a term that means that it is highly unlikely for the F(PN, PRIVACYKEY) hash function to return the same hash value for different privacy keys 34. Thus, it may be highly unlikely for a particular website 36 to use the F(PN, PRIVACYKEY) function (with its associated privacy key 34) to obtain the same hash value 32 for two different processor numbers 30. Thus, this feature ensures that it is highly likely for a particular web site 36 to identify each computer system with a different, unique processor number 30. Specification, pp. 5-6.

Yet another property of the F(PN, PRIVACYKEY) function (in some embodiments) may be that the F(PN, PRIVACYKEY) function is non-commutative, as described below:

F(F(PN,PRIVACYKEY),PRIVACYKEY')) ≠

F(F(PN,PRIVACYKEY'),PRIVACYKEY)),

where "PRIVACYKEY'" represents a privacy key 34 that is different from the privacy key 34 that is represented by "PRIVACYKEY." As a result of the non-commutative property, it may be very difficult to correlate the information that is associated with the computer system 10 (and user) on different databases (on different web sites 36) when different privacy keys 34 are used. Specification, p. 6.

Many different hash functions may be used, in various embodiments, that satisfy one, more than one, or all of the properties described below. For example, in some embodiments, a secure hash algorithm (SHA), an algorithm that satisfies all of the properties described above, may be used. Specification, p. 6.

In some embodiments, the computer system 10 may notify the user of the system 10 when a particular web site 36 is requesting system identification. For example, this notification may be in the form of a prompt in a window that is formed on a display 14 (see Fig. 3) of the computer system 10. In this manner, the user may either permit the web site 36 to obtain the identification (provided by the hash value 32) or reject the request. In some embodiments, the user may have an option to turn off the prompt. Specification, p. 6.

Besides prompting the user about the identification request, the computer system 10 may take measures to prevent a rogue web site 36 from submitting an incorrect privacy key 34 for purposes of obtaining a hash value 32 that is associated with another web site 36. For example, in some embodiments, the request for identification may involve a two-part identification procedure. First, the web site 36 sets the privacy key 34 by executing (if authorized, as described below) an instruction (called SETKEY(PRIVACYKEY)) of the processor 200 (see Fig. 2). Referring to Fig. 2, as described below, the SETKEY(PRIVACYKEY) function may be associated with ring zero (i.e., the highest level) of an operating system 28. As a result, the computer system 10 may not permit execution of this processor instruction until the computer system 10 validates the provided privacy key 34 by executing a software program called a driver 19.

After the privacy key 34 is validated by execution of the driver 19, the web site 36 may then be authorized to execute a processor instruction called HWID() (i.e., the HWID() instruction may not have an input parameter) that is associated with ring three (i.e., a lower privilege level) of the operating system 28 to obtain the hash value 32. Specification, pp. 6-7.

More particularly, in some embodiments, the above-described identification procedure may involve interaction between the operating system 28, an Internet browser 27 (Internet Explorer ® or Netscape Navigator ®, as examples) and the driver 19. For example, because the SETKEY(PRIVACYKEY) instruction is associated with ring zero, the web site 36 may not by itself cause execution of the instruction to obtain the hash value 32, as the web site 36 may only have access to ring three (a lower privilege level) and higher rings (i.e., even lower privilege levels) of the operating system 28. However, the driver 19 may have ring zero privileges and thus, may form a bridge between the web site 36 and the ring zero privileges of the operating system 28. In this manner, when the web site 36 attempts to execute SETKEY(PRIVACYKEY) instruction, the driver 19 may be called by the operating system 28 to cause the processor 200 to validate the privacy key 34 before providing the hash value 32. In the execution of the driver 19, the processor 200 may use results obtained from the execution of the browser 27 to validate the privacy key 34, as described below. Specification, p. 7.

Referring to Fig. 4, when executed by the processor 200, the driver 19 may cause the processor 200 to perform the following functions. In particular, the driver 19 may cause the processor 200 to determine (diamond 50) if the user has enabled an option to prompt the user when an identification request is received. If so, the processor 200 prompts (block 52) the user (via the display 14 (see Fig. 2), for example) that a web site 36 has submitted an identification request and waits for the user to indicate (via a keyboard 24 or move 26 (see Fig. 2), as examples) whether the user desires to reject the request. If so, the processor 200 rejects the request by notifying (block 56) the web site 36. Specification, pp. 7-8.

However, if the user did not reject the request, then the processor 200 may determine (diamond 58) whether the browser 27 is currently being executed. If so, the program 19 causes the processor 200 to communicate (block 60) the privacy key 34 to the browser 27 so that when the processor 200 executes the browser 27 (on another thread, for example), the processor 200 may compare the URL of the web site 32 to the privacy key 34. Subsequently, the processor 200, communicates the results of the comparison for use by the driver 19. In this manner, when the processor 200 subsequently executes the driver 19, the processor 200 determines (diamond 62) whether the privacy key 34 matches the URL of the web site 36. If not, the processor 200 rejects the request and notifies (block 56) the web site 36 about the rejection of the identification request. Otherwise, the processor 200 executes (block 64) the SETKEY(PRIVACYKEY) instruction to set the privacy key to be used for the encryption of the processor number 30. In this manner, the web site 36 that submitted the privacy key 34 may cause the processor 200 to execute the HWID() instruction to cause the processor 200 to produce an indication of the hash value 32. However, if the privacy key

34 has not been set, then the processor 200 returns an indication of an error rather than the indication of the hash value 32. Specification, p. 8.

Referring back to Fig. 3, in some embodiments, the computer system 10 may include a bridge, or memory hub 16. The processor 200 and the memory hub 16 may be coupled to a host bus 23. The memory hub 16 may provide interfaces to couple the host bus 23, a memory bus 29 and an Accelerated Graphics Port (AGP) bus 11 together. The AGP is described in detail in the Accelerated Graphics Port Interface Specification, Revision 1.0, published on July 31, 1996, by Intel Corporation of Santa Clara, California. The system memory 18 may be coupled to the memory bus 29, and store the driver 19, the browser 27 and portions of the operating system 28 (not shown in Fig. 3). A graphics accelerator 13 (that controls the display 14) may be coupled to the AGP bus 11. A hub communication link 15 may couple the memory hub 16 to another bridge circuit, or input/output (I/O) hub 20. Specification, pp. 8-9.

In some embodiments, the I/O hub 20 includes interfaces to an I/O expansion bus 25 and a Peripheral Component Interconnect (PCI) bus 21. The PCI Specification is available from The PCI Special Interest Group, Portland, Oregon 97214. A network interface 12 (a modem or a local area network (LAN) card, as examples) may be coupled to the PCI bus 21 and provide a communication path for the computer system 10 to communicate with the web sites 36. In this manner, the processor 200 may interact with the network interface 12 to communicate with the web sites 32. The I/O hub 20 may also include interfaces to a hard disk drive 37 and a CD-ROM drive 33, as examples. An I/O controller 17 may be coupled to the I/O expansion bus 25 and receive input data from the keyboard 24 and the mouse 26, as examples. The I/O controller 17 may also control operations of a floppy disk drive 22. Copies of the driver 19 may be stored on, as examples, the hard disk drive 32, a diskette or a CD-ROM, as just a few examples. Specification, p. 9.

Referring to Fig. 5, as an example, the processor 200 may include a bus interface unit (BIU) 208 that is coupled to address, control and data lines of the host bus 23 to, among other operations, retrieve instructions and data from the system memory 18. For the instructions, the processor 19 may include an instruction unit 203 that is coupled to the bus unit 208 to decode the instructions. In this manner, the instruction unit 203 may have buffers and a cache to store the instructions. A control unit 208 (of the processor 200) may receive signals from the instruction unit 203 that indicate the decoded instructions. For example, the signals may indicate the instruction to perform the SETKEY(PRIVACYKEY) function or the instruction to perform the HWID() function. Specification, p. 9.

In response to the instruction that is indicated by the instruction unit 203, in some embodiments, the control unit 208 may retrieve corresponding elementary instructions, called microcode, from a microcode read only memory (ROM) 210 of the processor 200 and execute the microcode. For example, microcode 211 to cause the processor 200 to perform the SETKEY(PRIVACYKEY) and HWID() instructions may be stored in a microcode read only memory (ROM) 210. In performing the execution of an instruction, the control unit 208 may control an arithmetic logic unit (ALU) 212, registers 214 and an addressing unit 206. Specification, pp. 9-10.

In other embodiments, the circuitry to perform the SETKEY(PRIVACYKEY) and HWID() instructions may be hardwired instead of being implemented in microcode. The processor number 30 may be replaced by another identifier that identifies the computer system 10. A privacy key other than a string that indicates an URL may be used. Applications other than applications being executed by web sites may request identification of the computer system 10. For example, other computer systems that are connected through a local area network (LAN) may request identification from the computer system 10. Specification, p. 10.

VI. ISSUES

A. **Can claims 1, 3-5, 21 and 22 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for independent claim 1?**

B. **Can claims 6, 8, 9, 23 and 24 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for independent claim 6?**

C. **Can claims 10-14, 25 and 26 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for independent claim 10?**

D. **Can claims 15-20 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for independent claim 15?**

E. **Can claims 27-30 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for independent claim 27?**

F. **Can claims 31-34 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for independent claim 31?**

G. **Can claims 35-38 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for independent claim 35?**

VII. GROUPING OF THE CLAIMS

Claims 1, 3-5, 21 and 22 can be grouped together; claims 6, 8, 9, 23 and 24 can be grouped together; claims 10-14, 25 and 26 can be grouped together; claims 15-20 can be grouped together; claims 27-30 can be grouped together; claims 31-34 can be grouped together; and claims 35-38 can be grouped together. With this grouping, all claims of a particular group stand or fall together. Furthermore, regardless of the grouping set forth by the Examiner's rejections, the claims of each group set forth in this section stand or fall together with respect to the other groups. In other words, any claim of a particular group set forth in this section does not stand or fall together with any claim of any other group set forth in this section.

VIII. ARGUMENT

All claims should be allowed over the cited references for the reasons set forth below.

A. Can claims 1, 3-5, 21 and 22 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for independent claim 1?

The method of claim 1 includes receiving a request from a first computer system for identification of a second computer system and retrieving a processor number that identifies a processor of the second computer system. The method includes encrypting the processor number with a key that is associated with the first computer system to produce a hash value. The hash value is provided to the first computer system in response to the request.

The Examiner rejects independent claim 1 under 35 U.S.C. § 103(a) in view of U.S. Patent No. 5,120,939 (herein called "Claus") and U.S. Patent No. 5,774,544 (herein called "Lee"). Claus is generally directed to a security system for a smart card. Lee is generally directed to a method and apparatus for encrypting and decrypting microprocessor serial numbers.

More specifically, Claus is directed to a security system in which a smart card provides a personal identification number to an authentication device. *See, for example*, Claus, 4:21-63. Claus neither teaches nor suggests that this personal identification number is somehow related to the identity of a processor of the disclosed smart card. Rather, the identification number is related to the identification of a *user* of the smart card, not to the identification of a *processor* of the smart card (emphasis added). Thus, the Examiner relies on the modification of Claus so that the personal identification number of Claus is replaced with Lee's microprocessor serial number. However, the Examiner provides no support for the alleged suggestion or motivation to modify Claus so that Claus' personal identification number is replaced with a microprocessor serial number. Such a suggestion or motivation for this modification must be present in the prior art to establish a *prima facie* case of obviousness. Furthermore, the Examiner must show, with specific citations, language from a prior art reference showing the alleged suggestion or motivation. However, the Examiner has failed to provide such support. *See, Ex parte Gambogi,* 62 USPQ2d 1209, 1212 (Bd. Pat. App. & Int. 2001); *In re*

Rijckaert, 28 USPQ2d 1955, 1957 (Fed. Cir. 1993); M.P.E.P. § 2143. As "obviousness cannot be predicated on what is unknown," a *prima facie* case of obviousness has not been established for independent claim 1. *In re Spormann*, 150 USPQ 449, 452 (CCPA 1966).

Additionally, the modification of Claus, as proposed by the Examiner, would render Claus' security system unsatisfactory for its intended purpose. More specifically, modifying Claus so that the personal identification number (provided by Claus' smart card) is replaced with a number that identifies a microprocessor of Claus' smart card would create a security system in which the smart card identifies its processor rather than identifying a user of the smart card. Thus, such a modification would create a smart card that no longer identifies the information stored on the smart card with a particular user. Therefore, for at least this additional, independent reason, there is no suggestion or motivation in the art to modify Claus as contended by the Examiner. *See* M.P.E.P. § 2145.X.D.

Claims 3-5, 21 and 22 are patentable for at least the reason that these claims depend from an allowable claim.

Thus, the § 103 rejections of claims 1, 3-5, 21 and 22 are improper and should be reversed.

B. Can claims 6, 8, 9, 23 and 24 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for independent claim 6?

The apparatus of claim 6 includes an interface that is adapted to receive a request from a computer system for identification of the apparatus and furnishing hash value that identifies the apparatus to the computer system. The apparatus includes a processor that is coupled to the interface and is adapted to encrypt a processor number that identifies the number with a key that is associated with the computer system to produce the hash value.

The Examiner rejects independent claim 6 under 35 U.S.C. § 103(a) in view of Claus and Lee. Claus teaches a smart card that provides a personal identification number, not a number that identifies a processor of the smart card. Furthermore, there is no teaching in Claus that the personal identification number is somehow correlated to the identity of the processor of the smart card. Thus, the Examiner relies on the modification of Claus with Lee's microprocessor serial number so that the personal identification number of Claus is replaced with this microprocessor serial number. However, the Examiner fails to show where the prior art contains the alleged suggestion or motivation to modify Claus in this manner. Furthermore, such a modification would render Claus' security system unsatisfactory for its intended purpose. In this manner, with this modification, the smart card of Claus would identify a serial number of its processor, rather than identify a user that uses the smart card. Such a modification, therefore, would not identify a user with the information that is stored on the smart card. *See,* M.P.E.P. § 2145.X.D.

Claims 8, 9, 23 and 24 are patentable for at least the reason that these claims depend from an allowable claim.

Thus, for at least the reasons set forth above, the § 103 rejections of claims 6, 8, 9, 23 and 24 are improper and should be reversed.

C. Can claims 10-14, 25 and 26 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for independent claim 10?

The article of claim 10 includes a storage medium that is readable by a first processor-based system. This storage medium stores instructions to cause a processor to receive a key from another processor-based system for identifying this other system. The instructions cause the processor to determine whether the key is valid and based on the identification, selectively authorizing encryption of an identifier that identifies the first system with the key to produce a hash value.

The Examiner rejects claims 10-14, 25 and 26 under 35 U.S.C. § 103(a) in view of U.S. Patent No. 5,825,884 (herein called "Zdepski") and Schneier, Bruce, *Applied Cryptography* (John Wiley & Sons 1996) (herein called "Schneier"). Zdepski generally teaches a server system for transferring subscriber information requests to information service providers. Schneier generally discloses cryptographic techniques.

The article of claim 10 recites that the instructions cause the processor to receive a key for identifying another processor-based system and based on this identification, selectively authorize encryption of an identifier. The Examiner contends that because Schneier somehow teaches an identity-based verification, Schneier teaches the selective authorization of claim 10. Final Office Action, 3. However, the Examiner does not point to any language in Schneier or Claus teaching selectively authorizing encryption based on identification of another processor-based system. Therefore, for at least the reason that the combination of references fails to teach or suggest all claim limitations, the Examiner fails to establish a *prima facie* case of obviousness for independent claim 10.

Claims 11-14, 25 and 26 are patentable for at least the reason that these claims depend from an allowable claim.

Thus, for at least the reasons set forth above, the § 103 rejections of claims 10-14, 25 and 26 are improper and should be reversed.

D. Can claims 15-20 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for independent claim 15?

Claim 15 recites a microprocessor that includes an instruction unit that is adapted to indicate when the instruction unit receives an instruction that requests an identifier that identifies the microprocessor. The microprocessor also includes an execution unit that is coupled to the instruction unit and is adapted to, in response to the indication from the instruction unit, encrypt a key with the identifier to produce a hash value. The microprocessor also includes a bus interface unit that is coupled to the execution unit and is adapted to furnish an indication of the hash value to external pins of the microprocessor.

The Examiner rejects independent claim 15 under 35 U.S.C. § 103(a) in view of the combination of Claus and Schneier. However, the Examiner fails to establish a *prima facie* case of obviousness for independent claim 15 due to the fact that the combination of Claus and Schneier does not contain several of the limitations of independent claim 15. For example, neither Claus nor Schneier teaches or suggests an instruction unit that indicates when the instruction unit receives an instruction that requests an identifier that identifies a microprocessor. Furthermore, neither Claus nor Schneier teaches or suggests an execution unit that is adapted to furnish an indication of a hash value to external pins of a microprocessor. Thus, as neither Claus nor Schneier teaches or suggests either the instruction unit or the execution unit of independent claim 15, a *prima facie* case of obviousness has not been established for this claim.

Claims 16-20 are patentable for at least the reason that these claims depend from an allowable claim.

Thus, the § 103 rejections of claims 15-20 are improper and should be reversed.

E. Can claims 27-30 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for independent claim 27?

The method of independent claim 27 includes providing a request to a second computer system for the second computer system to provide an identification of the second computer system. The method includes receiving a hash value from the second computer system. This hash value is generated by the encryption of a key that is associated with a first computer system with an identifier that identifies the second computer system. The method includes using the hash value to identify information associated with a user of the second computer system. The information is stored in a database that is maintained by the first computer system.

The Examiner rejects independent claim 27 under 35 U.S.C. § 103(a) in view of Claus and Schneier. However, the Examiner fails to show where either Claus or Schneier teaches or suggests a hash value that identifies information associated with a user of a second computer system, where this information is stored in a database that is maintained by a first computer system. Although the personal identification number of Claus may identify a person using the smart card and thus, may identify certain information stored on the smart card with information that is associated with this person, Claus neither teaches nor suggests using a hash value to identify this information, when the information is stored on another computer system. Schneier does not teach or suggest the missing claim limitations.

Claims 28-30 are patentable for at least the reason that these claims depend from an allowable claim.

Thus, the § 103 rejections of claims 27-30 are improper and should be reversed.

F. Can claims 31-34 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for independent claim 31?

The article of claim 31 includes a storage medium that is readable by a first processor-based system. The storage medium stores instructions to cause a processor of the first processor-based computer system to provide a request to a second computer system for the second computer system to provide an identification of the second computer system. The instructions cause the processor to receive a hash value from the second computer system. This hash value is generated by the encryption of a key that is associated with the first computer system with an identifier that identifies the second computer system. The instructions cause the processor to use the hash value identify information associated with a user of the second computer system. This information is stored in a database that is maintained by the first computer system.

The Examiner rejects independent claim 31 under 35 U.S.C. § 103(a) in view of Claus and Schneier. However, Claus fails to teach or suggest using a hash value to identify information associated with a user of a second computer system, for the scenario in which this information is stored in a database that is maintained by a first computer system. Although the smart card of Claus may generate a hash value that identifies a particular user, Claus neither teaches nor suggests using this hash value to identify the user information in a database that is maintained by a first computer system. In this manner, Claus neither teaches nor suggests that the authentication device nor any other computer system uses a hash value in this manner. Furthermore, Schneier fails to teach or suggest the missing claim limitations. Therefore, for at least this reason, a *prima facie* case of obviousness has not been established for independent claim 31.

Claims 32-34 are patentable for at least the reason that these claims depend from an allowable claim.

Thus, the § 103 rejections of claims 31=34 are improper and should be reversed.

G. Can claims 35-38 be rendered obvious when the Examiner has failed to establish a *prima facie* case of obviousness for independent claim 35?

The system of claim 35 includes a database and a first computer that is coupled to the database to provide a request to a second computer for the second computer to provide an identification of the second computer. The first computer receives a hash value from the second computer. This hash value is generated by encryption of a key that is associated with a first computer with an identifier that identifies a second computer.

The first computer uses the hash value to identify information associated with a user of the second computer. This information is stored in a database.

The Examiner rejects independent claim 35 under 35 U.S.C. § 103(a) in view of Claus and Schneier. However, neither Claus nor Schneier teaches nor suggests a computer to use a hash value (generated as specified in claim 35) to identify information associated with a user of a second computer, where this information is stored in a database that is coupled to the first computer. Furthermore, Schneier does not teach or suggest the missing claim limitations.

Therefore, for at least the reasons set forth above, the § 103 rejections of claims 35-38 are improper and should be reversed.

IX. CONCLUSION

Applicant requests that each of the final rejections be reversed and that the claims subject to this appeal be allowed to issue.

Respectfully submitted,

Date: July 9, 2003

Fred G. Pruner, Jr., Reg. No. 40,779
TROP, PRUNER & HU, P.C.
8554 Katy Freeway, Suite 100
Houston, TX 77024-1805
713/468-8880 [Phone]
713/468-8883 [Facsimile]

APPENDIX OF CLAIMS

The claims on appeal are:

1. A method comprising:

receiving a request from a first computer system for identification of a second computer system;

retrieving a processor number that identifies a processor of the second computer system;

encrypting the processor number with a key associated with the first computer system to produce a hash value; and

providing the hash value to the first computer system in response to the request.

3. The method of claim 1, further comprising:

executing a processor instruction; and

retrieving the number in response to the execution of the instruction.

4. The method of claim 1, further comprising:

receiving the key from the first computer system.

5. The method of claim 1, wherein the key indicates an address of a web site.

6. An apparatus comprising:

an interface adapted to:

receive a request from a computer system for identification of the apparatus, and

furnish a hash value that identifies the apparatus to the computer system; and

a processor coupled to the interface and adapted to:

encrypt a processor number that identifies the processor with a key associated with the computer system to produce the hash value.

8. The apparatus of claim 6, wherein the processor comprises:

a memory adapted to store microcode for performing the encryption; and

a control unit coupled to the memory and adapted to execute the microcode to perform the encryption.

9. The apparatus of claim 6, wherein the processor is further adapted to: interact with the interface to receive the key from the computer system.

10. An article comprising a storage medium readable by a first processor-based system, the storage medium storing instructions to cause a processor to:

receive a key from another processor-based system for identifying said another processor-based system,

determine whether the key is valid,

based on the identification, selectively authorize encryption of an identifier that identifies the first system with the key to produce a hash value.

11. The article of claim 10, the storage medium storing instructions to cause the processor to:

use an address of said another system to determine whether the key is valid.

12. The article of claim 11, wherein the key indicates an URL address.

13. The article of claim 10, the storage medium storing instructions to cause the processor to:

execute an instruction to cause the processor to subsequently use the key to produce the hash value.

14. The article of claim 10, wherein the identifier comprises a processor number.

15. A microprocessor comprising:

an instruction unit adapted to indicate when the instruction unit receives an instruction that requests an identifier that identifies the microprocessor;

an execution unit coupled to the instruction unit and adapted to, in response to the indication from the instruction unit, encrypt a key with the identifier to produce a hash value; and a bus interface unit coupled to the execution unit and adapted to furnish an indication of the hash value to external pins of the microprocessor.

16. The microprocessor of claim 15, wherein the execution unit comprises:

a control unit; and

a memory coupled to the control unit and storing microcode to cause the control unit to use the key and the identifier to produce the hash value.

17. The microprocessor of claim 15, wherein the identifier comprises a processor number.

18. The microprocessor of claim 15, wherein the execution unit is adapted to use a one way hash function to produce the hash value.

19. The microprocessor claim 15, wherein the execution unit is adapted to use a non-commutative hash function to produce the hash value.

20. The microprocessor of claim 15, wherein the execution unit is adapted to use a collision free hash function to produce the hash value.

21. The method of claim 1, wherein the processor number identifies a microprocessor of the second computer system.

22. The method of claim 21, wherein the processor number uniquely identifies the microprocessor.

23. The computer system of claim 6, wherein the processor number identifies a microprocessor of the apparatus.

24. The computer system of claim 23, wherein the processor number uniquely identifies the microprocessor.

25. The article of claim 14, wherein the processor number identifies a microprocessor of the first system.

26. The article of claim 25, wherein the processor number uniquely identifies the microprocessor.

27. A method comprising:

providing a request to a second computer system for the second computer system to provide an identification of the second computer system;

receiving a hash value from the second computer system, the hash value being generated by encryption of a key associated with a first computer system with an identifier that identifies the second computer system; and

using the hash value to identify information associated with a user of the second computer system, the information being stored in a database maintained by the first computer system.

28. The method of claim 27, wherein the identifier that identifies the second computer system comprises a processor number.

29. The method of claim 27, wherein the key indicates an address of a web site of the first computer system.

30. The method of claim 27, wherein the first computer system is located at a remote location relative to the second computer system.

31. An article comprising a storage medium readable by a first processor-based system, the storage medium storing instructions to cause a processor of the first processor-based computer system to:

provide a request to a second computer system for the second computer system to provide an identification of the second computer system;

receive a hash value from the second computer system, the hash value being generated by encryption of a key associated with the first computer system with an identifier that identifies the second computer system; and

using the hash value to identify information associated with a user of the second computer system, the information being stored in a database maintained by the first computer system.

32. The article of claim 31, wherein the identifier that identifies the second computer system comprises a processor number.

33. The article of claim 31, wherein the key indicates an address of a web site of the first computer system.

34. The article of claim 31, wherein the first computer system is located at a remote location relative to the second computer system.

35. A system comprising:

a database; and

a first computer coupled to the database to:

provide a request to a second computer for the second computer to provide an identification of the second computer,

receive a hash value from the second computer, the hash value being generated by encryption of a key associated with the first computer with an identifier that identifies the second computer, and

use the hash value to identify information associated with a user of the second computer, the information being stored in the database.

36. The system of claim 35, wherein the identifier that identifies the second computer comprises a processor number.

37. The system of claim 35, wherein the key indicates an address of a web site of the first computer.

38. The system of claim 35, wherein the first computer is located at a remote location relative to the second computer.